Token Bus Protocol Number 2

Patent Application

GB 2 280 572 B (application no. 9 315 722.0)

Mr Kim Lyon

Contact Details :-

Mr Kim Lyon
Suite 12397
72 New Bond St.
London
W1S 1RR
U.K.

Ph.:- +44-7785-362 319

Web Site :- wrldcomp.com

Token Bus Protocol Number 2

Abstract

The Token Bus Protocol Number 2 is a simple but sophisticated protocol that uses a numerical token that is passed retrospectively by the network hardware from one unit to another thus providing high speed data transfer through the resulting low data link overhead . The token is passed in an active manner and in a high speed passive manner . The protocol operates on a bussed topology and as such has the inherrent high reliability of this topology . In addition the protocol has an arbitrary unit number assignment system providing greatly increased ease of connectability of units . The protocol also has provision for the connection of intra and inter network gateways . The protocol can be implimented easily and cheaply .

The Token Bus Protocol Number 2

The Token Bus Protocol Number 2 , henceforth also refered to in this patent application as The Token Bus Protocol , has been designed to provide a simple but sophisticated solution to many of the problems found within current local area networking systems .

The Token Bus Protocol uses a numerical token that is passed by the network hardware from one unit to another thus providing high speed data transfer through the resulting low data link overhead . The data link overhead is approximately the same as the data link overhead with a Token Loop \ Ring network - typically around 3 % . This is as opposed typical data link overheads for Carrier Sense Multiple Access \ Collision Detection protocols of 25 % to 50 % . CSMA \ CD protocols are commonly used on bussed topologies . The data link overhead of CSMA \ CD protocols is high due to both the communication response time and the collisions and their associated random retry times . The data link overhead increases exponentially in proportion to the number of units trying to gain access to the network at any one time due to the effect of the collisions .

The Token Bus Protocol operates on a bussed topology and as such has the inherrent high reliability of this topology . The bus can be as simple as two wires , a wired ored electro - optical network or a radio based communication medium .

If one unit in a bus network breaks down the operation of the network as a whole is not effected . This is as opposed to the inherrent unreliability of the loop \ ring network where the reliability of the network is dependent on the reliability of each of the individual units on the network . If one unit on a loop \ ring network breaks down the whole network breaks down . As such the Mean Time Before Failure of the loop \ ring network is equal to the MTBF of the individual units divided by the number of units on the network .

The Bussed Topology

```
      --------------------------------------
     |            |            |            |
 --------     --------     --------     --------
| Unit 1 |   | Unit 2 |   | Unit 3 |   | Unit 4 |
 --------     --------     --------     --------
```

Figure 1

The Loop \ Ring Topology

```
     --------     --------     --------     --------
 -->| Unit 1 |-->| Unit 2 |-->| Unit 3 |-->| Unit 4 |->-
|    --------     --------     --------     --------    |
|                                                       |
 -------------------------------------------------------
```

Figure 2

The arrow shows the direction of communication from the transmitter to the receiver .

The Token Bus Protocol uses Active and Passive Retrospective Token Passing . Retrospective Token Passing minimises the ammount of required network activity to pass the token . Passive Token Passing provides an increase in speed over Active Token Passing by further reducing the ammount of network activity required to pass the token . Further the relative placement of the Token Take Up Slots ensures that equal priority for network access is maintained for all units .

The Token Bus Protocol , in addition , has an arbitrary unit number assignment system providing greatly increased ease of connectability of units . Units can be plugged into the network without setting up their addresses .

The Token Bus Protocol also has provision for the connection of intra and inter network gateways . This allows traffic to be easily managed on the network and allows for porting into other networks .

Conventions

The Following conventions are used :-

1) The most significant bit (the left most bit) shown is shifted out first .

2) All bit patterns are shown in binary except those shown with an H suffix which are shown in hexadecimal .

3) ... indicates as per above with the implied alterations as per the specified progressions .

4) ~~~ indicates an unspecified transmission whose content is not important to the example .

5) -> where used in transmission sequences is used to point to what next occurs in time .

The Token Bus Protocol on a Basic Level

The protocol shown following is an example of the implementation of the Token Bus Protocol .

The Token Bus Protocol is defined by the uniqueness of the Token Bus Protocol characters and message portions and their relationhips , usages and the subsequent protocol operation .

The way that the characters \ message portions are uniquely defined are determined by the actual protocol implementation .

For Example :- If the Token Bus Protocol uses 8 bit asychronous data communication the unique characters can be defined by using a preassigned range of bytes which are not used by the non unique characters . This does restrict the range of bytes that can be used in the data without the use of techniques such as byte stuffing but is useful in circumstances where only asychronous data communication is available .

If the Token Bus Protocol uses a known line state data communication system the unique characters are defined by the line state changes .

If the Token Bus Protocol uses a sychronous data communication system the unique characters are defined by their unique bit patterns .

The Key Aspects of the Token Bus Protocol

1) The message start is uniquely identified .

2) The destination address path is subsequently (by position) identified .

3) The source address path is subsequently (by position) identified .

4) The data is subsequently (by position) identified .

5) The message end is uniquely identified .

6) The message error check syndrome is correspondingly (by position) indentified .

An Example of The Token Bus Protocol on a Basic Level

The Token Bus Protocol as implemented in a synchronous data communication method uses the following basic format :-

1) All characters \ data have a basic length of 8 bits .

2) There are 3 unique characters :-

the Flag character with a bit pattern of 01111110 ,
the Poll character with a bit pattern of 01111111 ,
and the Terminate character with a bit pattern of 11111111 .

3) All non unique characters (ie. data) are checked on a bit by bit basis for the occurence of consecutive 1's .

   At the transmitter if 5 consecutive 1's occur then a 0 is inserted .

   IE. the bit pattern 11111111 becomes 111110111

   Similarly at the receiver if 5 consecutive 1's followed by a 0 occur then the 0 will be removed .

   IE. the bit pattern 11111000 becomes 1111100

   As such the uniqueness of the unique characters is preserved .

4) Addresses are encoded as non unique characters and consist of gateway numbers and unit numbers . The bit pattern or number group of the gateway numbers is uniquely different than the unit numbers . The gateway address path within a destination address preceedes the unit number . As such the unit number delimits the destination address .

   IE. if the gateway addresses are defined by the upper nibble being set to 1111 then the following example shows the use of gateway addresses within a destination address .

1111 0001	first gateway address
1111 0010	second gateway address
1111 0001	third gateway address
0001 0101	final destination unit address

Figure 3

   Source addresses are encoded in the same way and show the path taken from the destination to the source .

5) The error check syndrome is encoded as a non unique set of characters and may use any technique such as Hamming codes or Cyclic Redundantcy Check codes .

The Token Bus Protocol on a Message Level

The Token Bus protocol consists of two types of messages encoded in the following basic format :-

The first consists of :- FLAG
SOURCE ADDRESS
SOURCE ADDRESS
POLL

Figure 4

and is used to actively identify the presence of the unit and to indicate that no data is to be transfered .

The second consists of :- FLAG
DESTINATION ADDRESS
SOURCE ADDRESS
DATA
ERROR CHECK SYNDROME
POLL

Figure 5

and is used to transfer data .

Token Passing

Access to the network is passed onto a subsequent unit at the transmission of the poll . The token is passed retrospectively . As such the unit holding the token does not know the number (address) of the unit that it is passing the token on to . The unit that takes up the token knows the preceeding unit's number and hence can take up occupancy of the network when the token is passed into it's numerical and time domain .

The token is either passed in an active sequence where the subsequent unit takes up the token and passes it on without or with transmitting data or in a passive sequence where the subsequent unit either takes up the token if it has data to transmit or allows the token to pass onto a subsequent unit if it does not have data to transmit .

The Active Token Passing is used to identify all the units on the network when a new unit comes onto the network .

The Passive Token Passing is used during normal operation when no all unit identification is required . The Passive Token Passing having significantly less overhead than the Active Token Passing and hence effectively being the high speed token passing mode .

Active Token Passing

FLAG
~~~
SOURCE ADDRESS 1
~~~
POLL
inactive byte or bytes inserted if and as required
FLAG
~~~
SOURCE ADDRESS 2
~~~
POLL

Figure 6

The token position is tracked in the active token passing sequence by the use of the source address .

In the above example unit number 2 knows that unit number 1 is the immediately preceeding unit numerically and hence in actuallity and as such places it's transmission immediately after unit number 1 .

If a consecutive numbered unit is not on the network an inactive byte is inserted in between the poll and the subsequent flag . This becomes a free slot for a subsequent unit to occupy in the token passing sequence .

If a unit is switched on it will monitor the network for a specified time (eg. 65536 bytes time) for any activity . If there is no activity it will assign itself the lowest number (eg. 01H) and transmit a " no data to be transfered " message and then monitor again for any activity . If the network is active it will monitor for an inactive byte and will assign the number corresponding to that inactive byte to itself .

If a unit is disconnected from the network , on the next Active Token Passing Sequence it will be replaced by a number of inactive bytes corresponding to the number of units between the previous unit and the subsequent unit for one cycle and then the subsequent unit will place just one inactive byte between the previous unit and itself .

IE. if units 1 , 2 and 4 are present on the network and the active token passing mode is being used there will be no inactive bytes between units 1 and 2 and there will be one inactive byte between units 2 and 4 . If unit 2 is taken off the network there will be initially 2 inactive bytes between units 1 and 4 (corresponding to units 2 and 3) and then on the subsequent cycle there will be a single inactive byte between units 1 and 4 .

The inactive byte is used for the subsequent entering of other units into the token passing sequence .

The Terminate character is used to indicate the end of a transmission when an error condition occurs .

Passive Token Passing

Passive Token Passing consists of a Token Identification Sequence that consists of either a no data to transfer message or a data to transfer message that is followed by a Token Take Up Period that consists of slots during which subsequent units can take up the token and hence access the network .

```
FLAG
~~~
SOURCE ADDRESS n
~~~
POLL
unit number 0 Token Take Up Slot
unit number n+1 Token Take Up Slot
unit number n+2 Token Take Up Slot
unit number n+3 Token Take Up Slot
...
unit number m Token Take Up Slot
  wrap around
unit number 1 Token Take Up Slot
...
unit number n-1 Token Take Up Slot
```

Figure 7

where m and consequently the number of Token Take Up Slots corresponds to the maximum unit number on the network as defined at the last Active Token Passing sequence . This includes the unit number 0 Token Take Up Slot .

The token position is tracked in the Passive Token Passing Sequence by the use of the source address and the Token Take Up Slot . The Token Take Up Slot is defined by it's unique time position . The Token Take Up Slot has sufficient length to allow the transmission of the unit taking up the token to be identified as having commenced it's transmission . Typically this would be a two bit length to allow a 0 to 1 transition .

If a unit wishes to transmit it commences it's transmission within the Token Take Up Slot corresponding to it's number .

If no other unit transmits the last unit to transmit transmits an Active Token Passing Sequence without or with data being transfered (to identify and hence sychronise the take up slot positions) . This transmission immediately follows the last Token Take Up Slot and effectively corresponds to the unit number n Token Take Up Slot .

Initially Comming onto the Network

If a unit comes onto the network during Passive Token Passing it must first assign itself a free address . To do this it has to inform all other units on the network that it wishes for all units on the network to identify themselves . It does this by transmitting an Active Token Passing Sequence with a no data to transmit message with the source address set to 0 and consequently placed at the unit number 0 Token Take Up Slot . All other units then respond with a preset number of active token passing cycles . The unit uses the first cycle to identify a free address and then actively passes the token during the second cycle (and any subsequent cycles) to indicate it's presence on the network . Once the preset number of Active Token Passing Cycles has been completed all units will then revert to Passive Token Passing .

The placement of the unit number 0 Token Take Up Slot immediately following the unit Token Identification Sequence allows the unit number 0 token take up slot to be clearly identified without any reference to the number of units on the network . This means that the unit can gain access to the network immediately after one transmission rather than having to wait for at least two transmissions before being able to identify the unit number 0 Token Take Up Slot .

Addresses

The following address types are used :-

1) A unique number is reserved as a local network general broadcast number . This allows all units within a network to be simultaneously addressed .

2) A range of unique numbers are reserved for the unit numbers .

3) A range of unique numbers are reserved for global broadcast numbers . These allow party conversations to occur .

4) A range of unique numbers are reserved for gateway unit numbers . These allow communication with the gateways to occur .

5) A unique gateway general broadcast number . This allows all gateways to be simultaneously addressed .

6) A range of unique numbers are reserved for the gateway numbers . These allow the data to be transfered through the gateway .

7) A unique number is used as a wide area network general broadcast number . This allows broadcasts to all units on the network to occur .

Please Note :- implementations of the protocol may assign a zero range of numbers to any of the above addresses . IE. there may be no numbers assigned to a specific address type .

The destination address for units transfering data through gateways consists of path information specifying the path to be taken from the source unit to the destination unit .

The source address for units transfering data through gateways consists of path information specifying the path that was taken from the source unit to the destination unit .

Each gateway strips off the destination address corresponding to itself and adds the source address corresponding to itself as it passes the message from it's network receiver on one network to it's network transmitter on the other network .

Examples of the Operation of the Token Bus Protocol

The convention used is that the arrow points to what occurs next in time .

A Single Unit is Present On the Network

```
    ---<---
   |       |
  FLAG     |
   01H     | 65,536 bytes inactive time
   01H     |
  POLL     |
   |       |
    --->---
```

Figure 8

A Second Unit is Plugged into the Network

The second unit hears the first unit during its' inactive time . It triggers an Active Token Passing sequence and then assigns itself the next free number -> 2 and places itself into the cycle .

```
    ---<---
   |       |
  FLAG     |
   01H     |
   01H     |
  POLL     |
  FLAG     | triggers active token passing
   00H     |
   00H     |
  POLL     |
   |       |
    --->---
```

Figure 9

Second Cycle

```
     ---<---
    |       |
  FLAG      |
   01H      | 254 bytes inactive time
   01H      |
  POLL      |
  FLAG      |
   02H      |
   02H      |
  POLL      |
    |       |
     --->---
```

Figure 10

Subsequent Cycles

```
     ---<---
    |       |
  FLAG      |
   01H      | 1 byte inactive time
   01H      |
  POLL      |
  FLAG      |
   02H      |
   02H      |
  POLL      |
    |       |
     --->---
```

Figure 11

Subsequent Passive Token Passing Sequence

After a preset number of Active Token Passing Cycles the network goes into Passive Token Passing . Unit number 2 (the last unit to transmit) undertakes the Token Identification Sequence .

```
     ---<---
    |       |
  FLAG      | 2 Token Take Up Slots
   02H      | - unit number 0
   02H      | - unit number 1
  POLL      |
    |       |
     --->---
```

Figure 12

A Third Unit is Plugged into the Network

```
     ---<---
    |       |
  FLAG      |
   02H      |
   02H      |
  POLL      |
  FLAG      | triggers active token passing sequence
   00H      |
   00H      |
  POLL      |
  FLAG      |
   01H      |
   01H      |
  POLL      |
    |       |
     --->---
```

Figure 13

Second Cycle

```
     ---<---
    |       |
  FLAG      |
   01H      | 253 bytes inactive time
   01H      |
  POLL      |
  FLAG      |
   02H      |
   02H      |
  POLL      |
  FLAG      |
   03H      |
   03H      |
  POLL      |
    |       |
     --->---
```

Figure 14

Subsequent Cycles

```
     ---<---
    |       |
  FLAG      | 1 byte inactive time
   01H      |
   01H      |
  POLL      |
  FLAG      |
   02H      |
   02H      |
  POLL      |
  FLAG      |
   03H      |
   03H      |
  POLL      |
    |       |
     --->---
```

Figure 15

Passive Token Passing

```
     ---<---
    |       |
  FLAG      | 3 Token Take Up Slots
   03H      | - unit number 0
   03H      | - unit number 1
  POLL      | - unit number 2
    |       |
     --->---
```

Figure 16

The First Unit is Unplugged from the Network

The Next Active Token Passing Cycle

```
     ---<---
    |       |
  FLAG      | 254 bytes inactive time
   02H      |
   02H      |
  POLL      |
  FLAG      |
   03H      |
   03H      |
  POLL      |
    |       |
     --->---
```

Figure 17

Subsequent Cycles

```
     ---<---
    |       |
  FLAG      |
   02H      | 1 byte inactive time
   02H      |
  POLL      |
  FLAG      |
   03H      |
   03H      |
  POLL      |
    |       |
     --->---
```

Figure 18

Communication

```
      --------------------------------------
     |            |            |            |
 --------     --------     --------     --------
| Unit 1 |   | Unit 2 |   | Unit 3 |   | Unit 4 |
 --------     --------     --------     --------
```

Figure 19

If unit no. 1 wishes to transfer data to unit no. 3 it transmitts the sequence :-

```
  FLAG                    start of packet
   03H                    destination address
   01H                    source address
  DATA                    data
  ERROR CHECK SYNDROME    error check syndrome
  POLL                    end of packet
```

Figure 20

Usage of Gateways

```
                    0F0H  ---------  0FEH
      -------------------| Gateway |---------------------
     |           |       ---------          |           |
 --------    --------                   --------    --------
| Unit 1 |  | Unit 2 |                 | Unit 1 |  | Unit 2 |
 --------    --------                   ---------   --------
```

Figure 21

If unit no. 1 in the left hand network wishes to transfer data to unit no. 2 in the right hand network it transmitts the sequence :-

```
FLAG                  start of packet
0F0H                  gateway address   { destination
 02H                  destination unit  { address path
 01H                  source address
DATA                  data
ERROR CHECK SYNDROME  error check syndrome
POLL                  end of packet
```

Figure 22

After passing through the gateway the packet becomes :-

```
FLAG                  start of packet
 02H                  destination address
0FEH                  gateway address  { source
 01H                  source address   { address path
DATA                  data
ERROR CHECK SYNDROME  error check syndrome
POLL                  end of packet
```

Figure 23

0FEH,01H being the complete source address .

0FEH being the address of the unit (in this example a gateway) that last transmitted the message . This address also being the one that is used to signify the current token position and hence to provide a link to the next unit in the numerical order .

The What If's

The What If's are a series of questions and answers designed to address the practical issues of the Token Bus Protocol implementation .

Abbreviations :- Q) Question
A) Answer
E) Example

The What If's with regards to the Active Token Passing

Q) What if a message is corrupted but the flag and poll have been received correctly ?

A) The unit maintains a count of the number of units currently on line and notices that the previous message should have come from the previous unit and acts as if the message had been received intact .

E) Given the numerical unit sequence :-

```
     01   message intact
     02   message intact
     06   message corrupted
 >   09
```

Unit 9 is counting down from it's own number . The transmission of unit 6's corrupted message triggers off it's own subsequent message transmission .

Q) But suppose that :-

```
     01   message intact
     02   message corrupted
     06   message corrupted
 >   09
```

occurs ?

A) The counter is still effective .

Q) But suppose that :-

```
     01   message intact
     02   message corrupted
     03   message corrupted - new unit
 >   06
 >   09
```

occurs ? Doesn't the counter lose sync ?

A) Yes unit 6 and unit 9 will collide and both will detect an error . Both will cease their transmission and unit 1 will then take over .

Q) What if a message is corrupted , the flag has been received correctly but the poll has been corrupted ?

A) The subsequent unit will wait for a total of 512 bytes (maximum message length) for the unit to cease transmission and will then transmit a message .

Q) In the unlikely event that not all units have received a corrupted message what would happen then ?

A) The 512 byte wait would consist of the actual message followed by inactive bytes . The units that have received the message correctly would then count through these inactive bytes as if the subsequent units had gone off line . When the first unit had reached it's respective position it would transmit it's message and the token passing sequence would resume .

Q) Would this not upset the token passing on the next cycle when the other units resume transmission ?

A) No . An inactive byte will be present after the unit whose message was corrupted . The subsequent unit will slot in here and the other units will follow .

Q) But suppose the following occurs ?

01 message intact
02 message intact to 01 but corrupted to 03 , 04 and 05
03
04
06

A) Yes units 6 and 1 will collide because unit 1 thinks unit 2 (which is subsequently modified to unit 3 and then unit 4) is the previous unit . As such both units 6 and 1 will insert an inactive byte after unit 4 and will then simultaneuosly transmit a message and will collide . Unit 2 will see it as a non previous unit message and will then put in inactive bytes between unit 4 and itself . It will then transmit a message and the token passing sequence will be restored .

Q) What happens if the flag is corrupted ?

A) The subsequent unit still knows that it's turn is comming up . When the poll is transmitted it will take over .

Q) And if the poll is also corrupted ?

A) The subsequent unit waits the total of 512 bytes from the start of the message and then transmitts it's message .

Q) Does the unit recognise a corrupted flag as an active or as an inactive byte ?

A) It recognises it as an active byte and as such ensures that it doesn't transmit in the middle of a message .

Q) Doesn't this cause any problems with the recognising of inactive bytes ?

A) No because an active byte was expected . An inactive byte will be either all 0's or all 1's . Any other pattern indicates transitions on the line and hence activity .

Q) But given the practicalities of the implementation how can miss-timings and glitches at the start of the byte period be prevented from being confused ?

A) Yes there is a potential for confusion and hence there will be a 2 bit window for these to die down before the inactive byte sampling occurs . IE. there will be only 3.125 % of bit patterns that will produce an inactive byte look alike .

Q) What about the flag - that only needs 1 bit (the last 0) to be read as a 1 and then it is perceived as an inactive byte . Suppose this occurs ?

```
     01   message intact
     02   flag read as 01111111
 >   06
```

A) Unit 6 will know that it's previous unit has not transmitted and will insert inactive bytes . If during this period it notices activity it will then wait for the poll or it will wait for 512 bytes or another unit to transmit a message .

Q) But given all this doesn't it effect the way the unit goes off line ?

A) No because when it goes off line it notices that there is no activity and hence reverts to the 8192 byte wait before transmitting a message .

Q) Ok. So what registers are there ?

A) The state register that stores the current state of the
  off line \ address assigned \ message started state .
The currently transmitting unit's address
  - token position .
The previous unit's address .
The current unit's address .
The number of units on the network .
The 512 \ 8192 byte wait timer .

The mailbox function will be implemented in software .

Q) And how will the Token Position Register work ?

A) It will advance at the reception of a poll to the next unit number . It will also advance at the reception of any subsequent consecutive inactive bytes to subsequent unit numbers . When a message has been received intact it will use the source address contained in the message to determine the next unit number . It will require a qualifying register to store the source address and to allow the necessary decisions to be set up .

Q) And what about the Number of Units on the Network Register ?

A) That will be set up on each complete intact cycle .

The What If's with regards to the Passive Token Passing

Q) What if the last transmission is received corruptly ?

A) Any subsequent units that wish to access the network wait until the last unit retransmitts in order to achieve sychronisation .

Q) What if the current token passing unit is disconnected from the network ?

A) After the Token Identification Sequence and the Token Take Up Period and a subsequent period corresponding to the Take Up Slot of the current token passing unit a subsequent Token Take Up Period occurs during which the next subsequent unit takes over the Token Identification Sequence .

Q) What if a subsequent unit that wishes to access the network is disconnected from the network ?

A) After the Token Identification Sequence and the Token Take Up Period and a subsequent period corresponding to the Take Up Slot of the current token passing unit a subsequent Token Take Up Period occurs during which the subsequent unit takes over the Token Identification Sequence .

Claims

1) A token bus protocol that uses

   i) a bussed network topology . The Bussed Network Topology being by definition a single communication medium onto which all the network units are connected .

   and

   ii) a retrospective token passing method where each unit monitors the prior transmission sequence using this to identify the transmission of the previous unit and the number (address) of the previous unit and hence determining the starting position of it's transmission .

   The Retrospectivite Token Passing being where the current unit does not know the number (address) of the subsequent unit but where the current unit does know the number (address) of the previous unit . Thus knowledge is not maintained as to where the token is going to but is retained as to where the token has come from .

   and

   iii) Active Token Passing where each unit possesing the token allows the token to pass onto a subsequent unit by concluding it's transmission .

   The unit that holds the numerical token by definition has sole access to the network bus . The unit releases it's sole access to the network bus thus allowing the next unit to access the network by completing it's transmission .
   The next unit not having a specific number or necessarily a consecutive number but being the next numerically numbered unit currently connected to the network bus . The current unit indicates that it has the token by transmitting it's source address and releases it's sole access to the network bus by transmitting the poll (message end) character .

The unit's transmission occurs immediately after it's previous unit has finished transmission and with or without inactive bytes inserted between the previous unit's transmission and the unit's transmission as required .

and

iv) Passive Token Passing where after the Token Identification Sequence of the last unit to transmit their follows a Token Take Up Period during which subsequent units can take up the token and hence access the network .

The Token Take Up Period consisting of a series of inactive time slots . The first of which is to allow units comming onto the network to trigger an active token passing sequence in order to identify a free unit number . The subsequent slots corresponding to the units sequential to the last unit to transmit . The slots being wrapped around . If a subsequent unit whishes to access the network it commences it's transmission within it's corresponding time slot . Where no subsequent unit transmitts and on reaching slot of the last unit to transmit the last unit to transmit re-transmits the Token Identification Sequence and the cycle is recommenced .

In such a manner the token is taken up by the last unit to transmit and is then released to the network where it is passed passively from unit time slot to unit time slot without any transmission occuring until a unit whishes to take up the token .

2) A token bus protocol as defined in claim 1 that provides for Arbitrary Unit Number (Address) Assignment .

The Arbitrary Unit Number (Address) Assignment by definition being that the unit does not have a predefined number (address) but assigns itself an available number (address) on connection to the network bus .

The arbitrary unit number (address) assignment being accomplished by the use of the retrospective token passing in conjunction with the inactive bytes such that the unit comming onto the network assigns the number (address) corresponding to the source address plus 1 of the first received message that is immediately followed by an inactive byte . This being the first available address .

3) A token bus protocol as defined in claim 1 that provides for the connection to and communication via intra and inter network gateways .

The gateways being by definition a method of connecting the network to other networks of the same or differing types .

This being accomplished by the token bus protocol via destination and source address paths that have defined characteristics and defined positions and hence both the paths and their component addresses have identifyable positions .